5-1964

Mfour crystallographic fourier summation program

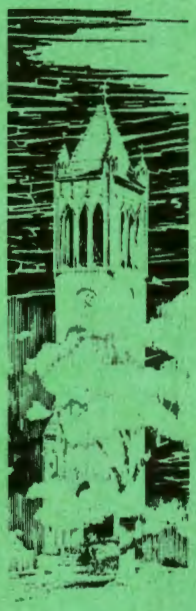
Mads Ledet
Iowa State University

Mfour crystallographic fourier summation program

Abstract
This program was written as a substitute for MIFRl. There are four notable differences: i) no symmetry manipulations are performed; ii) no sorting is made with regard to sine-cosine codes; iii) summation is done completely in each direction before proceeding; iv) program is for the IBM 7074 (20K).

Disciplines
Computer Sciences | Mathematics

IS-876



IOWA STATE UNIVERSITY

MFOUR
CRYSTALLOGRAPHIC FOURIER
SUMMATION PROGRAM

by

Mads Ledet

AMES LABORATORY

RESEARCH AND DEVELOPMENT REPORT

U.S.A.E.C.

IS-876

Mathematics and Computers (UC-32)
TID-4500, April 1, 1964

UNITED STATES ATOMIC ENERGY COMMISSION

Research and Development Report

MFOUR
CRYSTALLOGRAPHIC FOURIER SUMMATION PROGRAM

by

Mads Ledet

May, 1964

Ames Laboratory
at
Iowa State University of Science and Technology
F. H. Spedding, Director
Contract W-7405 eng-82

This report is distributed according to the category Mathematics and Computers (UC-32) as listed in TID-4500, April 1, 1964.

Printed in USA. Price $ 1.25 . Available from the

Office of Technical Services
U. S. Department of Commerce
Washington 25, D. C.

CONTENTS

MFOUR CRYSTALLOGRAPHIC FOURIER SUMMATION PROGRAM

Mads Ledet

1. ABSTRACT

This program was written as a substitute for MIFR1.[1] There are four notable differences: i) no symmetry manipulations are performed; ii) no sorting is made with regard to sine-cosine codes; iii) summation is done completely in each direction before proceeding; iv) program is for the IBM 7074 (20K).

2. GENERAL INFORMATION

2A. General Program Information

The need for a program of this type became evident when Iowa State made the transition to the IBM 7074. Several good programs have been written in Fortran but they have all been very specific programs designed for a particular crystal. It is hoped that this program is not only efficient but also general enough to be useful.

Input will be accepted in either Autocoder ten (10) card image blocks or Fortran single "card image" records. There is no practical limit to the number of input records, i.e., up to 10^8 records can be accepted. However the upper limit on the size of the Miller indices is presently set at 50.

---

1. W. G. Sly, and D. P. Shoemaker, "MIFR1: Two and Three-Dimensional Crystallographic Fourier Summation Program for the IBM 704 Computer", Office of Ordnance Research (Contract DA-19-020-Ord - 4696 National Institutes of Health Research Grant A2400 (July, 1960).

Output is in the approximate shape of a square. Four significant digits are retained with high order zeros deleted. The smallest interval available is 1/160 of a cube with integral multiples up to and including 8/160 with 20 points or less per layer in either direction.

The following summation scheme is used:

$$\rho(x, y, z) = \sum_{l} \left\{ \sum_{k} \left[\sum_{h} (Fcalc_0 \cos 2\pi hx + Fcalc_1 \sin 2\pi hx) \cos 2\pi ky + \sum_{h} (Fcalc_2 \sin 2\pi hx + Fcalc_3 \cos 2\pi hx) \sin 2\pi ky \right] \cos 2\pi lz + \sum_{k} \left[\sum_{h} (Fcalc_4 \cos 2\pi hx + Fcalc_5 \sin 2\pi hx) \sin 2\pi ky + \sum_{h} (Fcalc_7 \sin 2\pi hx + Fcalc_6 \cos 2\pi hx) \cos 2\pi ky \right] \sin 2\pi lz \right\}.$$

2B. Efficiency of Program

There are three things the user can do to obtain maximum efficiency in the running of this program.

1. Because it is physically impossible to use two tape drives on one channel simultaneously the tape option control card should specify the data input on the channel not used by the X summation output. The Y summation output should be on the same channel as the data input. With these conventions the tape input-output will never be on the same channel for either the X or the Y summations. The Z summation output tape is not used until all other tapes are finished so does not enter into the above considerations.

2. The output tapes are split at approximately T/2 where T is the total number of blocks on the data tape. This allows one output tape to rewind to be ready for input into the next phase while the second output tape is being written. Obviously "T/2" should be such that both output tapes are of similar length. This can be accomplished by having "T/2" be a number such that approximately 1/2 of the "L" values are in the preceeding input blocks, i.e., if L goes from 0 to 10 and the value of L = 5 refers to the blocks 800 to 882 then $800 \leq$ "T/2" ≤ 882 882. (Note the total number of blocks may be equal to or greater than 883 in this example.)

3. After each complete XYZ summation, the Fourier program will go back only as many summations as is necessary to obtain the next XYZ summation. This means that the origin control cards should be arranged so that the repetitive summations are eliminated.

   For example the following arrangement is the most efficient possible with these cards:

1.	X = 0	Y = 0	Z = 0	1/4	1/4	1/8
2.	X = 0	Y = 0	Z = 20	1/4	1/4	1/8
3.	X = 0	Y = 20	Z = 0	1/4	1/4	1/8
4.	X = 0	Y = 20	Z = 20	1/4	1/4	1/8
5.	X = 20	Y = 0	Z = 0	1/4	1/4	1/8
6.	X = 20	Y = 0	Z = 20	1/4	1/4	1/8

   With the preceding arrangement, the Fourier program will do only two X summations (1 and 5); three Y summations (1, 3 & 5); and six Z summations. In general, the Z summation should change each time with the X summation changing the least of all three.

3. OPERATING PROCEDURES

3A. Data Input

All reflections must be on a special tape that has been prepared beforehand.

There are two types of input:

1. AUTOCODER - data must be blocked ten (10) card images per block.

2. FORTRAN - data must be in single "card image" records. Everything beyond the eighth (8) word in each record will be ignored. The first eight (8) words must be identical to one (1) AUTOCODER card image.

The format of the input is as follows:

WORD#	COLUMNS		EXAMPLE
1	1-10	H	0000000010^{+}
2	11-20	K	0000000006^{+}
3	21-30	L	0000000009^{+}
4	31-40	Fcalc	0000687300$^{+(-)}$
5	41-50	F-code	0000000003^{+}
6	51-60		0000000000^{+}
7	61-70		0000000000^{+}
8	71-80		0000000000^{+}

Note that all words except #4 must have a plus sign zone punch in the units position. Also note that all input is right justified in the respective words. Fcalc will be treated as an automatic decimal number and so the implied decimal must remain fixed for all Fcalc's.

Input must be arranged according to the following procedure:

1. L indices in increasing sequence.

2. K indices in increasing sequence under each L.

3. H indices in increasing sequence under each K.

The input tape must end with a tape mark (End of File) and tape marks may not be used at any other point. Fortran segment marks will be ignored.

The last Autocoder block should be padded with zero cards (0000000000) to obtain a complete block of ten cards.

Word #5 has a special meaning for the program and refers to the particular summation Fcalc belongs to as follows:

F. Code	X	Y	Z
0	cos	cos	cos
1	sin	cos	cos
2	sin	sin	cos
3	cos	sin	cos
4	cos	sin	sin
5	sin	sin	sin
6	cos	cos	sin
7	sin	cos	sin

3B. Control Card Input

The following control cards must be used in the given sequence. Only one control card of each type except #4 (Origin Control) may be present in one run.

#1. Type of input:

The program will accept both FORTRAN and AUTOCODER. This card specifies which type of data input to expect.

1 (a)		1 (b)	
Columns		Columns	
0- 7	FOURIER	0- 7	FOURIER
11-17	FORTRAN	11-19	AUTOCODER
18-80	blank	20-80	blank

#2. Header label:

This information will be printed on every sheet of output.

Columns	
0- 7	FOURIER
11-80	Header information

#3. Tape options:

This card specifies which tapes are to be used for all phases of the program. The last entry refers to the length of the input tape and is usually near T/2 where T is the total number of input blocks. This entry is necessary and makes the program more efficient.

Columns	
0- 7	FOURIER
11-15	TAPES
21-22	1st data input
26-27	Alternate data input
31-32	Xsum output
36-37	Alternate Xsum output
41-42	Ysum output
46-47	Alternate Ysum output
51-52	Final output (23 is standard)
56-57	Alternate final output (23 is standard)
61-65	T/2 (approximately)

#4. Origin control:

This card determines the final output:

Columns	
0- 7	FOURIER
11-16	ORIGIN
21	X
23-25	Origin for X (Mult. of 1/160).
26	P
29-30	Number of X points (20)
31	Y
33-35	Origin for Y (Mult. of 1/160)
36	P
39-40	Number of Y points (20)
41	Z
43-45	Origin for Z (Mult. of 1/160)
46	P
49-50	Number of Z points (20)
51-54	MAXH
55-56	Value of largest of H, K, and L.
58-60	Length of X edge (1/8, 1/4, 3/8 1/2, 5/8, 3/4, 7/8, 1/1)
61-62	SF
63	+ positive power of 10 - negative power of 10
64	Power of 10

The scale factor (columns 63-64) should not be greater than 10-I where I is the largest number of digits in any Fcalc. The scale factor is merely a shift left (SL3) or a shift right and round (SRR3) so that too large a shift to the left will cause high order digits to be lost.

3C. Typical Run

A typical run will have the following cards.

1. START ACCNT
2. Remarks to operator about data input tape
3. XZLOAD TAPE 14
4. FORN
5. Fourier program
6. FORF
7. AUTOCODER control card (1 only)
8. HEADER control card (1 only)
9. TAPES control card (1 only)
10. ORIGIN control cards (variable number)
11. WTM
12. END ACCNT

Item 11 must be present so that the Fourier program can recognize the end of the control card deck.

PROGRAM ANALYSIS GUIDE

4A. Introduction

The following analysis guide consists of flow charts and explanations. The guide is fairly complete and should serve to clear up any questions as to what can be done.

4B. General Program Flow Diagram

BLOCK #

G001 Initialize program control switches.

G002 If first pass go to block G003.

G003 Get one-time-only control cards and use these to set up program flow. These control cards are the first three read and are related to input and output controls.

G004 This includes the SINCOS routine and various other checks to determine amount of computation needed.

G005 Go to correct phase.

G006 This is the X summation phase.

G007 Get X sum input data. On EOF the output is finished and the switch for this phase is turned off.

G008 This is the Y summation phase.

G009 Get Y sum input. This input is the X sum output. On EOF a check is made to determine if there is only one input tape. Output is finished and the switch for this phase is turned off.

G010 This is the Z summation phase.

G011 Get Z sum input. See G009.

G012 This is the routine that computes and stores the results for each phase. This routine is modified every time a new phase is entered since the table size and make-up varies.

G013 Branch to correct phase.

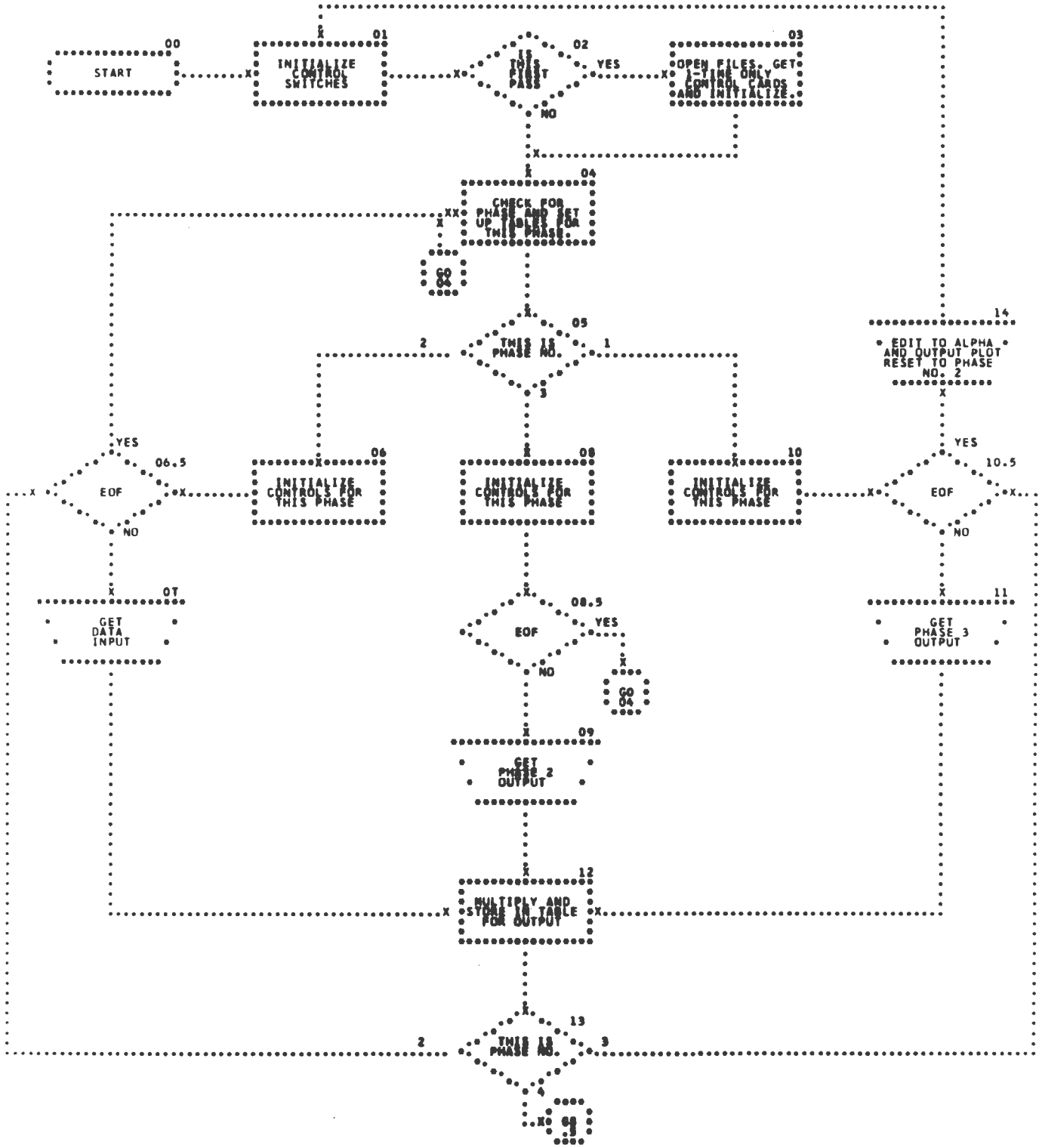
G014 Z summation is finished. Edit output into correct form, output tables, and go back to start program again.

4C. Initialization Phase Flow Diagrams

BLOCK #

H001 This block initializes various electronic switches to control program flow. The switches are as follows:

a. XSW: Controls access to Phase 2 (XSUM).
b. YSW: Controls access to Phase 3 (YSUM).
c. ZSW: Controls access to Phase 4 (ZSUM).
d. CHNGTP: Controls splitting of Phase 2 output tapes to prevent waiting for Phase 3 input.
e. CHNGTPE: Does same for Phase 3 as CHNTP does for Phase 2.


00
START
01
INITIALIZE CONTROL SWITCHES
02
IS THIS FIRST PASS
YES
NO
03
OPEN FILES, GET 1-TIME ONLY CONTROL CARDS AND INITIALIZE
04
CHECK FOR PHASE AND SET UP TABLES FOR THIS PHASE
GO 04
05
THIS IS PHASE NO.
1
2
3
06
INITIALIZE CONTROLS FOR THIS PHASE
08
INITIALIZE CONTROLS FOR THIS PHASE
10
INITIALIZE CONTROLS FOR THIS PHASE
14
EDIT TO ALPHA AND OUTPUT PLOT RESET TO PHASE NO. 2
06.5
EOF
YES
NO
07
GET DATA INPUT
08.5
EOF
YES
NO
GO 04
09
GET PHASE 2 OUTPUT
10.5
EOF
YES
NO
11
GET PHASE 3 OUTPUT
12
MULTIPLY AND STORE IN TABLE FOR OUTPUT
13
THIS IS PHASE NO.
2
3
4
GO 04

f. T1FULL: Controls output from Table 1. Switch on signifies table not completely empty.
g. T2FULL: Same control for Table 2 as T1FULL.
h. T2: Controls which table will be written out in case last table empty (T1FULL or T2FULL off).
i. CARDEND: Signals end of Phase 2 input (data input tape).

H002 A branch switch for one time initialization. Two switches are turned off:

a. FORT: On signifies FORTRAN data input.
b. ZOPEN: Controls error messages output by preventing attempts to use final output before tape file is opened.

H003 Index word ZERO will be used throughout program for zeroing areas.

H004 The control card tape file (HEADLAB) is opened.

H005 A control card is obtained.

H006 A check is made to determine if this card is a control card. If this is a control card, a branch is made to C1. Otherwise, go to H007.

H007 Is the final output tape file open so that error messages can be written?

H008 Yes, write error message.

H009 No, type message on console log.

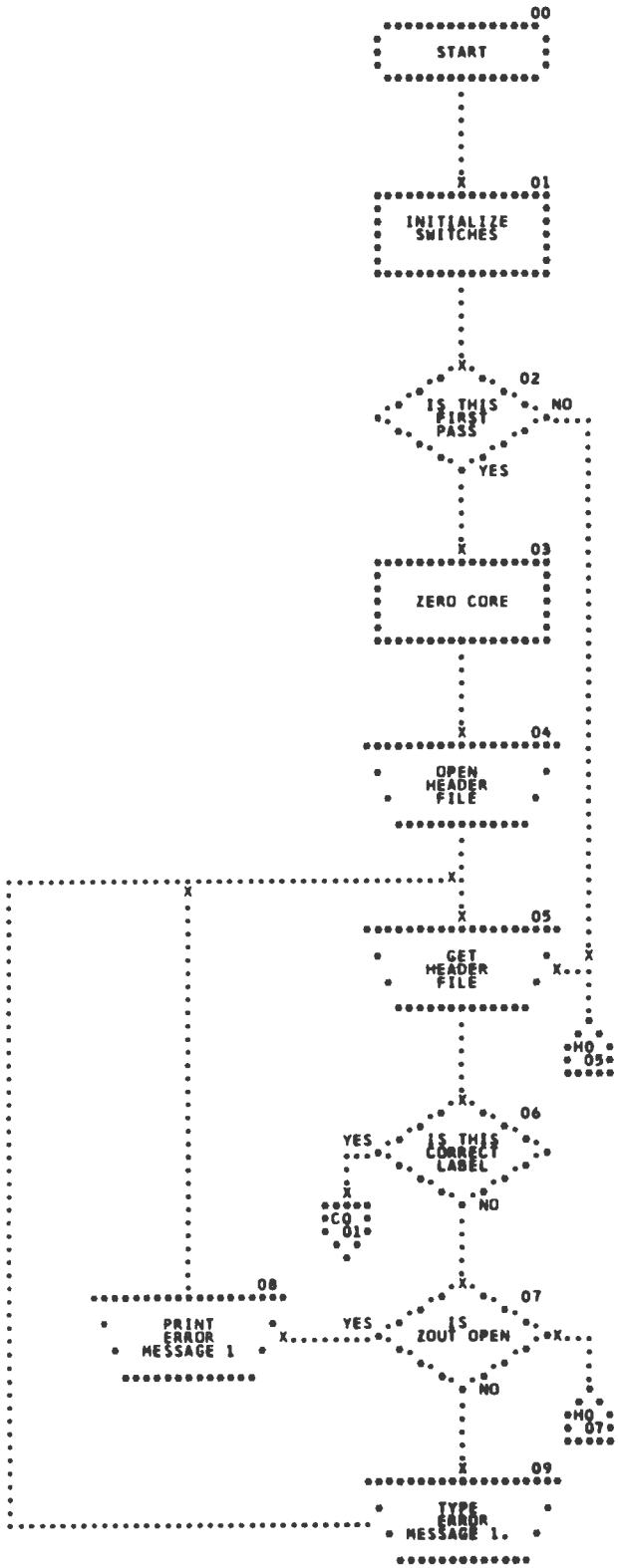
4D. Control Card Phase Flow Diagrams

BLOCK #

C001 A block of three branch switches that are set as each control
C002 card is read.
C003

C004 Expecting an ORIGIN control card to determine type of output from Phase 4.

C005 Data input tape file is opened. Two files are available depending on type of input.


00
START
01
INITIALIZE SWITCHES
02
IS THIS FIRST PASS
NO
YES
03
ZERO CORE
04
OPEN HEADER FILE
05
GET HEADER FILE
HO 05
06
IS THIS CORRECT LABEL
YES
NO
CO 01
07
IS ZOUT OPEN
YES
NO
HO 07
08
PRINT ERROR MESSAGE 1
09
TYPE ERROR MESSAGE 1.

C006 1st control card determines type of input. If input is FORTRAN the GET routines for data are modified to refer to the FORTRAN input file.

C007 Modify GETs.

C008 2nd control card determines output label. Columns 21-80 are moved without processing to the output label area (LABELAREA).

C009 3rd control card specifies channel and unit number for all tapes except HEADLAB. A check is made to determine if Basetape and Alt-tape channels are the same, if not, a warning message is given on console log and the Alt-tape channel is ignored. If input is FORTRAN, the XIN file must be last DTF. Since five files are left to be initiated, the last pass must be modified for only one file.

C010 XCHAN and YCHAN are BCB's for buffering of output. These are set for correct channels.

C011 LCOUNT is an index register that counts the number of data input records to determine when to split Phase 2 output. This splitting eliminates rewind waiting time. Since AUTOCODER is blocked 10 records TPLNGTH must be multiplied by 10 to give number of records.

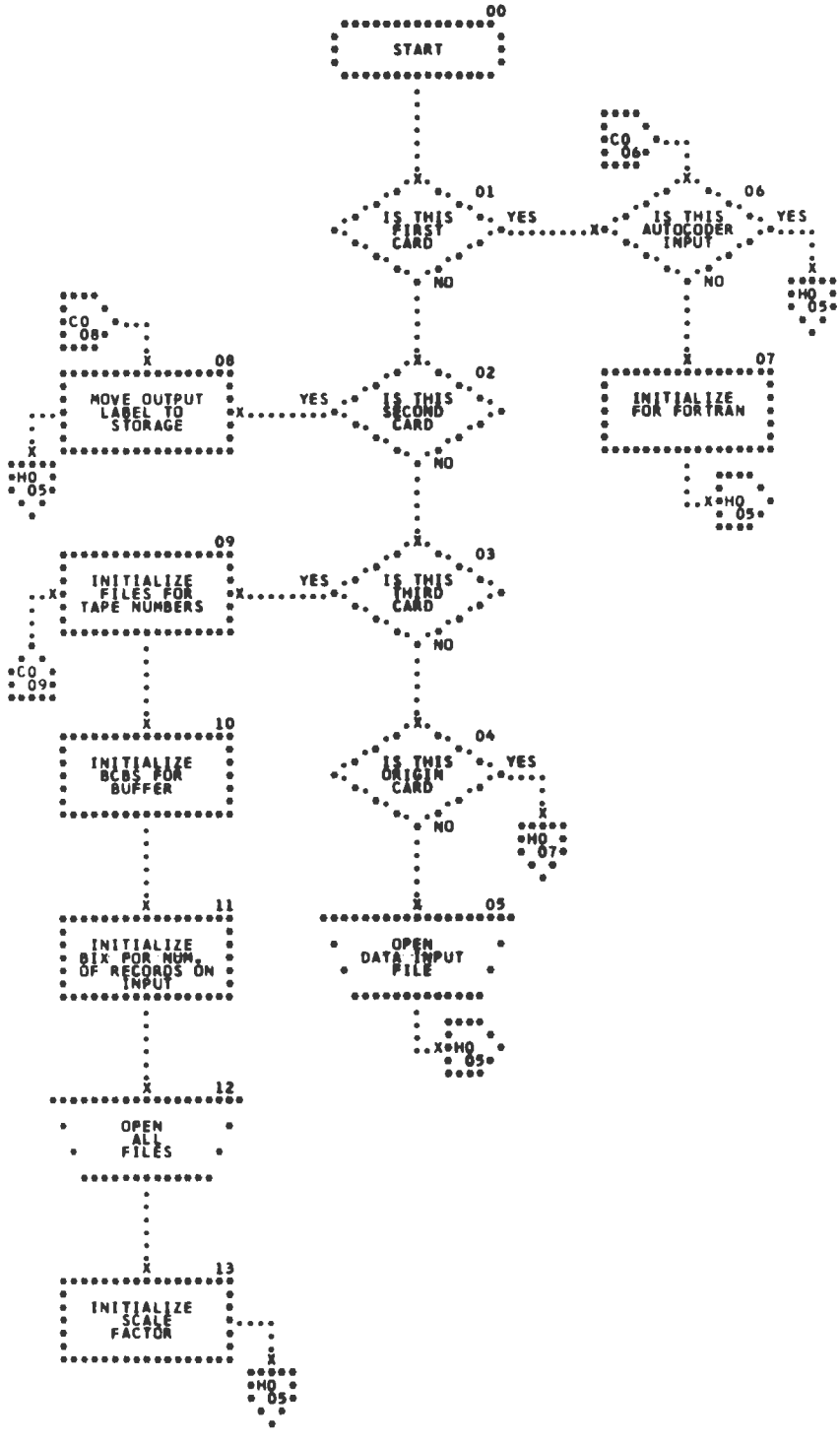
C012 Open all files since channels and units are specified. Set ZOPEN switch on.

C013 Scaling is accomplished by multiplying by 10^{w} where w is ± an integer. If SCALSGN is negative, the operation will be SRR3 while if SCALSGN is plus the operation will be SL3. If the power of 10 (w) is zero, the scaling operation is bypassed by setting the bypass branch.

4E. Initialization For-Correct-Phase Flow Diagrams

BLOCK #

I001 Accumulators are loaded with the parameters for Phase 2, 3 or 4 according to index register PASCNT1. SAMESNCOS is an electronic switch that controls the entrance to the SINCOS routine. If the new starting point and cube edge is same as for last summation SAMESNCOS is on.


00
START
01
IS THIS FIRST CARD
YES
NO
06
IS THIS AUTOCODER INPUT
YES
NO
CO 06
HO 05
02
IS THIS SECOND CARD
YES
NO
08
MOVE OUTPUT LABEL TO STORAGE
CO 08
HO 05
07
INITIALIZE FOR FORTRAN
HO 05
03
IS THIS THIRD CARD
YES
NO
09
INITIALIZE FILES FOR TAPE NUMBERS
CO 09
04
IS THIS ORIGIN CARD
YES
NO
HO 07
10
INITIALIZE BCBS FOR BUFFER
05
OPEN DATA INPUT FILE
HO 05
11
INITIALIZE DIX FOR NUM. OF RECORDS ON INPUT
12
OPEN ALL FILES
13
INITIALIZE SCALE FACTOR
HO 05

I002 If this is Phase 2, we are starting a new summation so a check must be made as to how many summations are equivalent to the last XYZ summations. This allows time to be saved by using each summation as many times as is possible before changing.

I003 If SINCOS table needed is same as old table, do not re-compute.

I004 If this is Phase 2 of 1st XYZ summation go directly to SINCOS routine because no table present.

I005 Compare last XYZ summation to new XYZ summation.

I006 Is last Phase 2 equal to new Phase 2.

I007 Is last Phase 3 equal to new Phase 3.

I008 Is last Phase 4 equal to new Phase 4.

I009 If all three summations are the same, ignore this control card and print warning message.

I010 Maximum number of points in any direction is 20.

I011 If last phase used less than 20 points, the multiplication routine has been modified by a branch so restore routine.

I012 Modify multiplication routine if number of points is less than 20 for this phase. Also modify ADD TO STORAGE (ASI) for this phase.

I013 Go to correct phase.

I014 Go to correct phase.

I015 Go to correct phase.

I016 Halt as phase switches are all off.

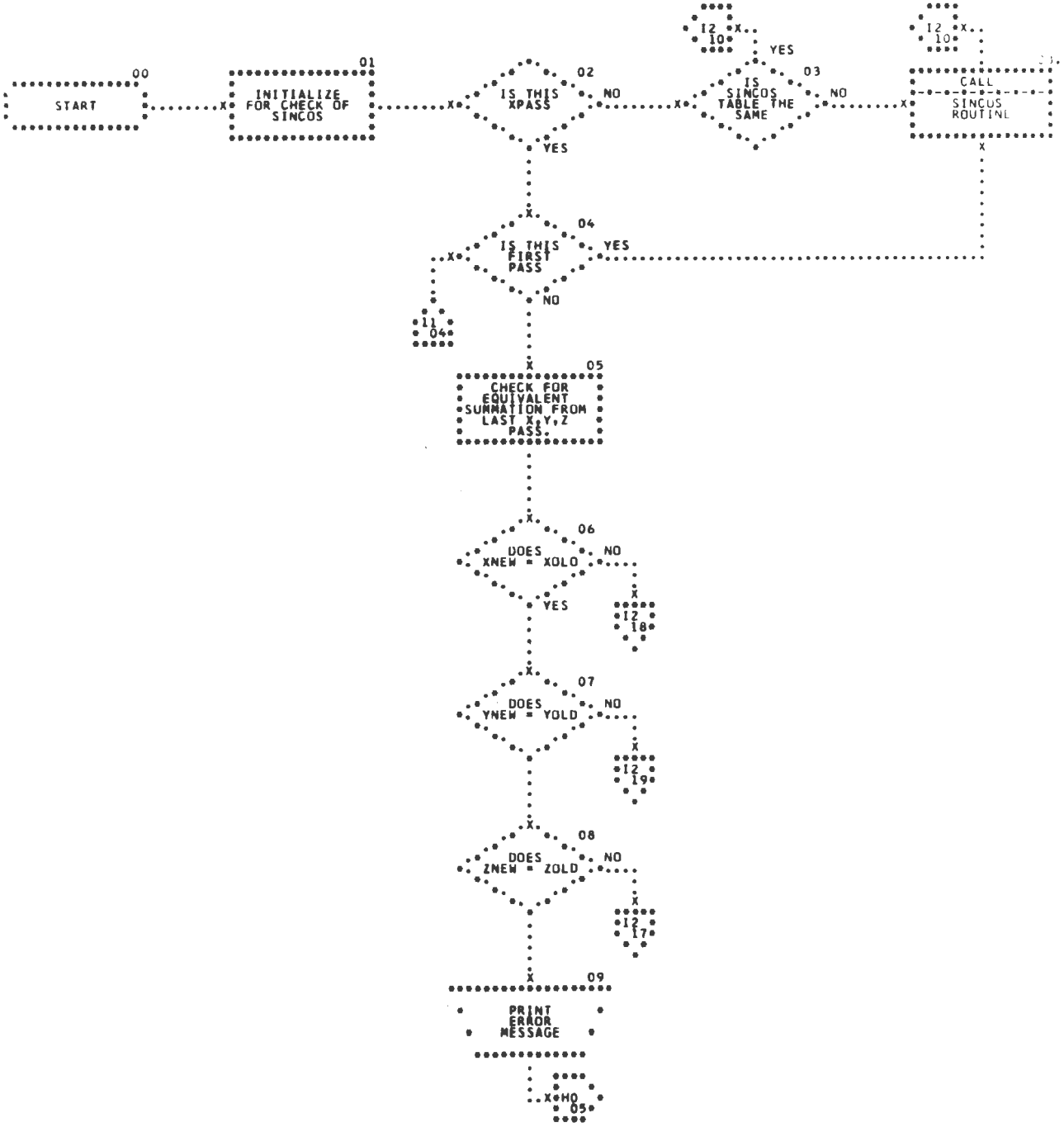
I017 Only Phase 4 must be recomputed.

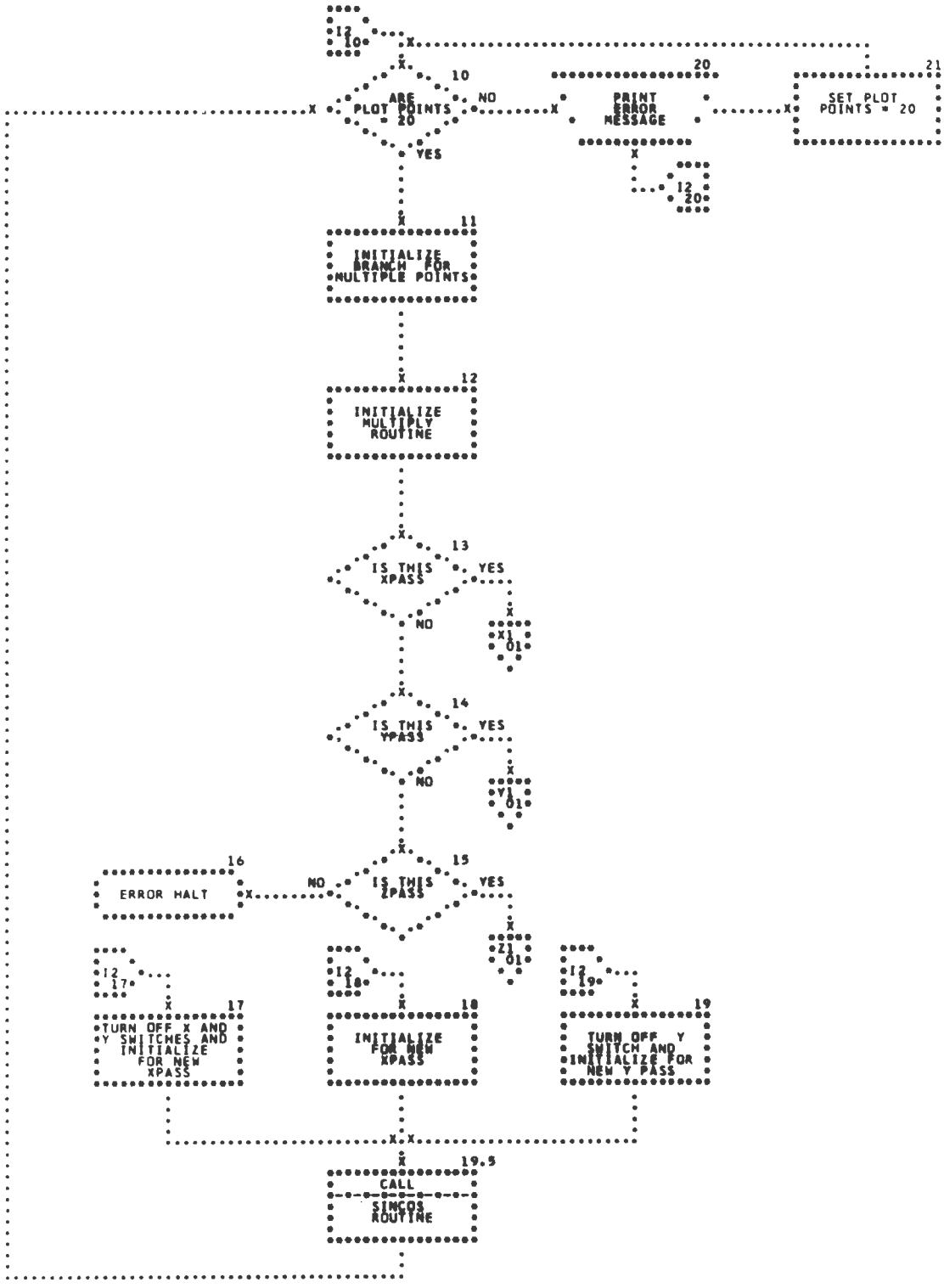
I018 All three phases must be recomputed.

I019 All but Phase 2 must be recomputed.

I020 Write warning about number of points requested.

I021 See number of points equal to 20.


00
START
01
INITIALIZE FOR CHECK OF SINCOS
02
IS THIS XPASS
NO
YES
I2 X I0
YES
03
IS SINCOS TABLE THE SAME
NO
I2 X I0
CALL SINCUS ROUTINE
04
IS THIS FIRST PASS
YES
NO
I1 04
05
CHECK FOR EQUIVALENT SUMMATION FROM LAST X,Y,Z PASS.
06
DOES XNEW = XOLD
NO
YES
I2 I8
07
DOES YNEW = YOLD
NO
I2 I9
08
DOES ZNEW = ZOLD
NO
I2 I7
09
PRINT ERROR MESSAGE
H0 05


12
10
10
ARE PLOT POINTS = 20
NO
20
PRINT ERROR MESSAGE
21
SET PLOT POINTS = 20
12
20
YES
11
INITIALIZE BRANCH FOR MULTIPLE POINTS
12
INITIALIZE MULTIPLY ROUTINE
13
IS THIS XPASS
YES
NO
X1
01
14
IS THIS YPASS
YES
NO
Y1
01
16
ERROR HALT
NO
15
IS THIS ZPASS
YES
Z1
01
12
17
17
TURN OFF X AND Y SWITCHES AND INITIALIZE FOR NEW XPASS
12
18
18
INITIALIZE FOR NEW XPASS
12
19
19
TURN OFF Y SWITCH AND INITIALIZE FOR NEW Y PASS
19.5
CALL SINCOS ROUTINE

4F. X-Summation Phase (#2) Flow Diagrams

BLOCK #

X001 Data input is brought into storage. The specific input file was determined in Block C006 and the GET was modified accordingly.

X002 The two index registers for zeroing tables are initiated. RECDCNT is a branch switch that is set after half the records have been processed. LCOUNT counts the records as they are processed. Index register TX is loaded with an RDW identical to the one specifying the first DA for X summation except that the RDW is plus and ignores the last two locations which are used for information transfer between phases.

X003 ACC3 is loaded with Fcalc.

X004 A branch switch that is on if no scale factor is needed.

X005 Scaling is obtained by shifting left or right obtaining an equivalence of powers of 10.

X006 Index register INDX will refer to the SINCOS table entry equal to sin 2πH X or cos 2πH X where $0 \leq i \leq$ XPTS. Input is in the form of +00000000HH.

X007 A branch switch that is on if no output from a X summation table is waiting. This eliminates several time consuming checks if they are not needed. With average to large amounts of data the last table will be completely written on tape long before a new table is ready.

X008 If output is waiting is the output channel busy. If so, no attempt is made to write on the output tape. This allows the program to do a maximum number of calculations for each tape write.

X009 Electronic switch T1FULL is checked to see if Table 1 is being written out. If not, Table 2 must be full since output is waiting.

X010 Write out one record from Table 1.

X011 Have all 21 records been written on tape.

X012 Zero Table 1 and initialize for new summation for use when Table 2 is full.

BLOCK #

X013 If output is to be split at this time CHNGTP, an electronic switch, is on. CHNGTP is first turned on when LCOUNT (block X002) has counted half of the records on the data tape. This allows one output tape to rewind before it is needed for Phase 3.

X014 Same for Table 2 as Block X010.

X015 Same for Table 2 as Block X011.

X016 Same for Table 2 as Block X012.

X017 Same for Table 2 as Block X013. Output tape can be signaled for splitting from either table.

X018 F-code is checked to determine two things:
a. Should INDX refer to the sine or cosine table.
b. Which entry in X sum table should TX refer to. See XBLD DA for more information.

X019 After multiplication we want TX to refer to the first entry under the particular value of K with which we are working.

X020 Force X sum output onto alternate tape. Also turn off ONEXTP to signify that there are two X sum output tapes. Branch back to continue the program.

X021 This is the return from the multiplication routine. OLDK and OLDL are needed to save K and L during GET.

X022 GET a new input card.

X023 Restore K and L.

X024 This is a branch switch that is off if T/2 records have not been read.

X025 Count the input records.

X026 T/2 records have been read. Turn on switch in Block X024. Set switch to split output at the next change in L.

X027 If the new L is smaller do not accept this record since the block for the new L has already been written on tape.

X028 If the new L is larger prepare to write out a block. Do not check K since it belongs in new table.

BLOCK #

X029 If the new L is equal check the new K to see if it is smaller. If so do not accept this record since it cannot be entered into table.

X030 If the new K is larger, prepare to store the old K in the table directly below the four table entries it refers to.

X031 The new K is equal. Restore TX to refer to the first entry for this K. This is necessary since the initialization in Block X018 is dependent on this procedure. Return to Block X003 to initialize for new Fcalc.

X032 Store old K and old L in the table directly below the four entries for old K. An electronic switch BRBACK determines whether or not we write out this table. KSAME is a branch switch that is set off if the new K is larger than old K. This allows us to store the old K in the table.

X033 Is BRBACK off? If so prepare to write out this table.

X034 Store record marks to signal the end of each record in the table. There are as many records as there will be X points in the final plot.

X035 If Table #1 is not empty, go to finish writing.

X036 If Table #2 is not empty, go to finish writing.

X037 If T2 is on, we know that Table #1 was written out last and we must now write out Table #2. This switch is necessary because many times the last table is empty and the table switch is off so we can not determine which is the new table.

X038 T2 is off so go to start writing out Table #1.

X039 We must now store entires in Table #2 so initialize TX for Table #2.

X040 If data input tape is empty, we must not go back to get a new Fcalc. Instead, finish writing out old table and go to Y summation phase.

X041 If the table is empty, go to set up for Y phase.

X042 T2 is on so write out Table #2.

X043 See block X040. We do not initialize TX since we can branch to beginning of routine and load it there.

BLOCK #

X044 See block X041.

X045 Finish writing out all the records remaining in this table since we must use this table for the summation because Table #1 is full.

X046 Zero table and set T2FULL off since table is empty.

X047 Must we split the output tape here?

X048 Same as block X045 except for Table #1.

X049 Same as block X046 except for Table #1.

X050 Same as block X047 except for Table #1.

X051 If we have only one output tape ONEXTP is on so close the file to prevent rotation of tapes. Otherwise, rotate tapes and zero core. Also set XSW off since we are done with this phase.

4G. Y-Summation Phase (#3) Flow Diagrams

BLOCK #

Y001 Initialize the two index registers used for zeroing the tables. Five asterisks are stored at the beginning of each table to signal the start of a new table when reading the input for the Z-sum. YLPX controls the input by branching to the correct routine for each of the four entries for each K. (See the DAs for more information.)

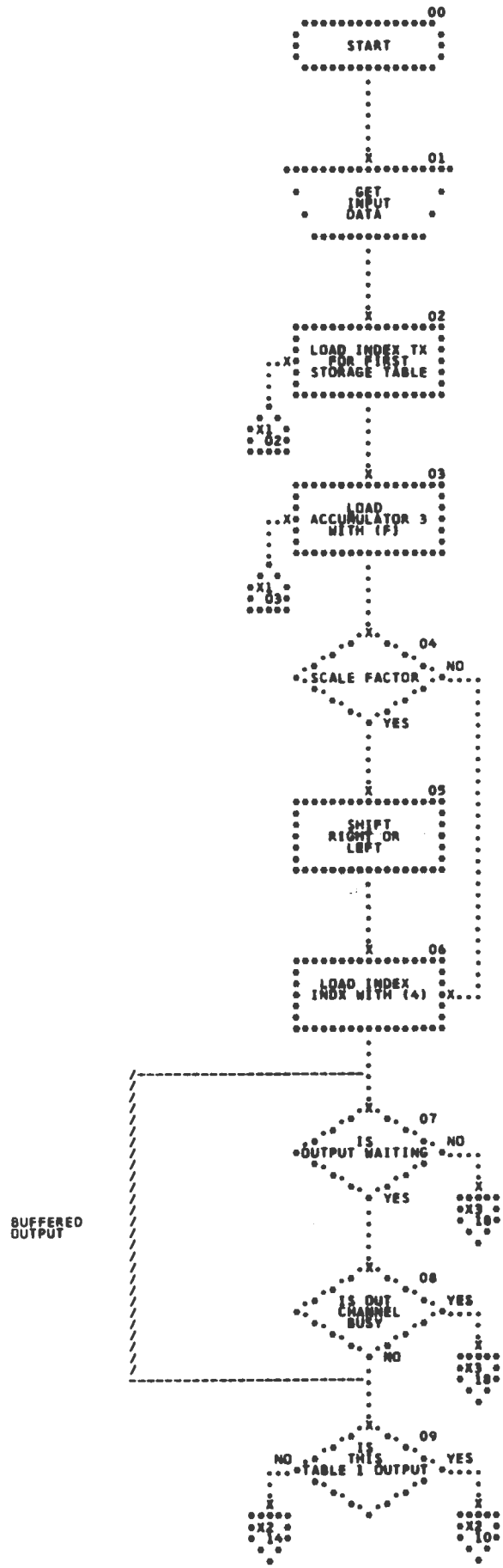
Y002 The first record is read into core. TX is not incremented since it already refers to the first table entry.

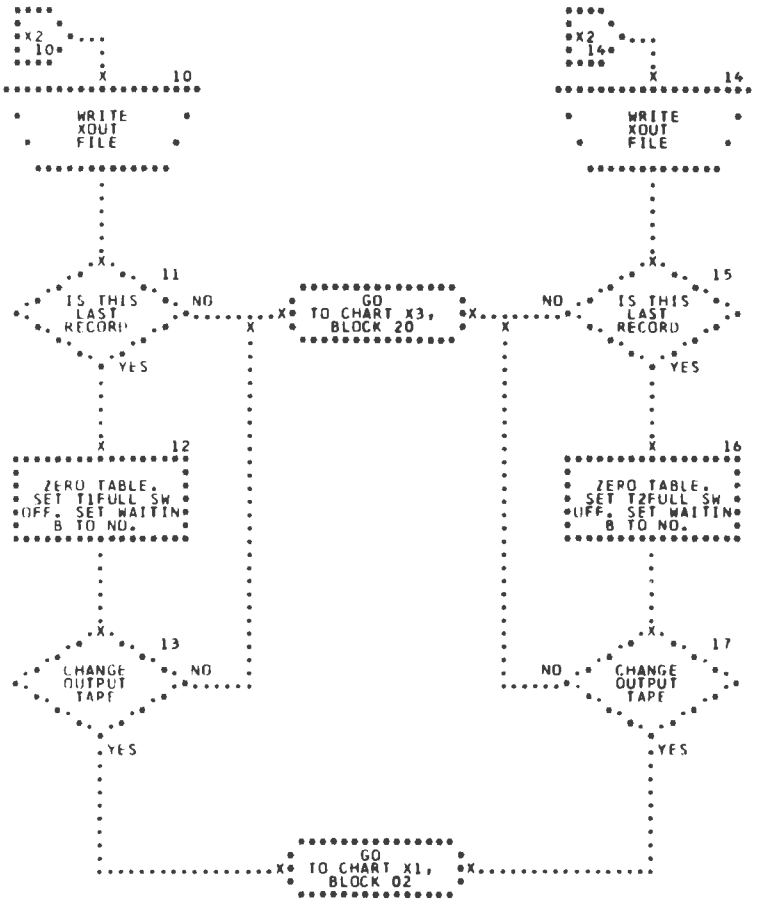
Y003 Acc 3 and INDX are loaded as for the X sum. Acc 3 will be loaded with four consecutive entries for each value of K that is loaded into INDX.

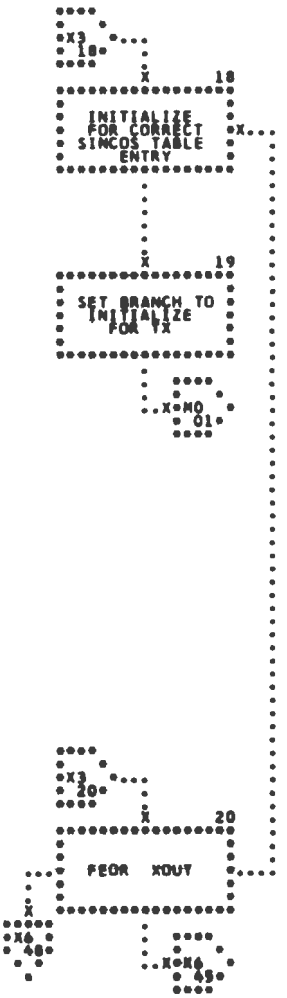
Y004 If Acc 3 is zero, do not process. For certain summations only one of the four entries will not be zero.

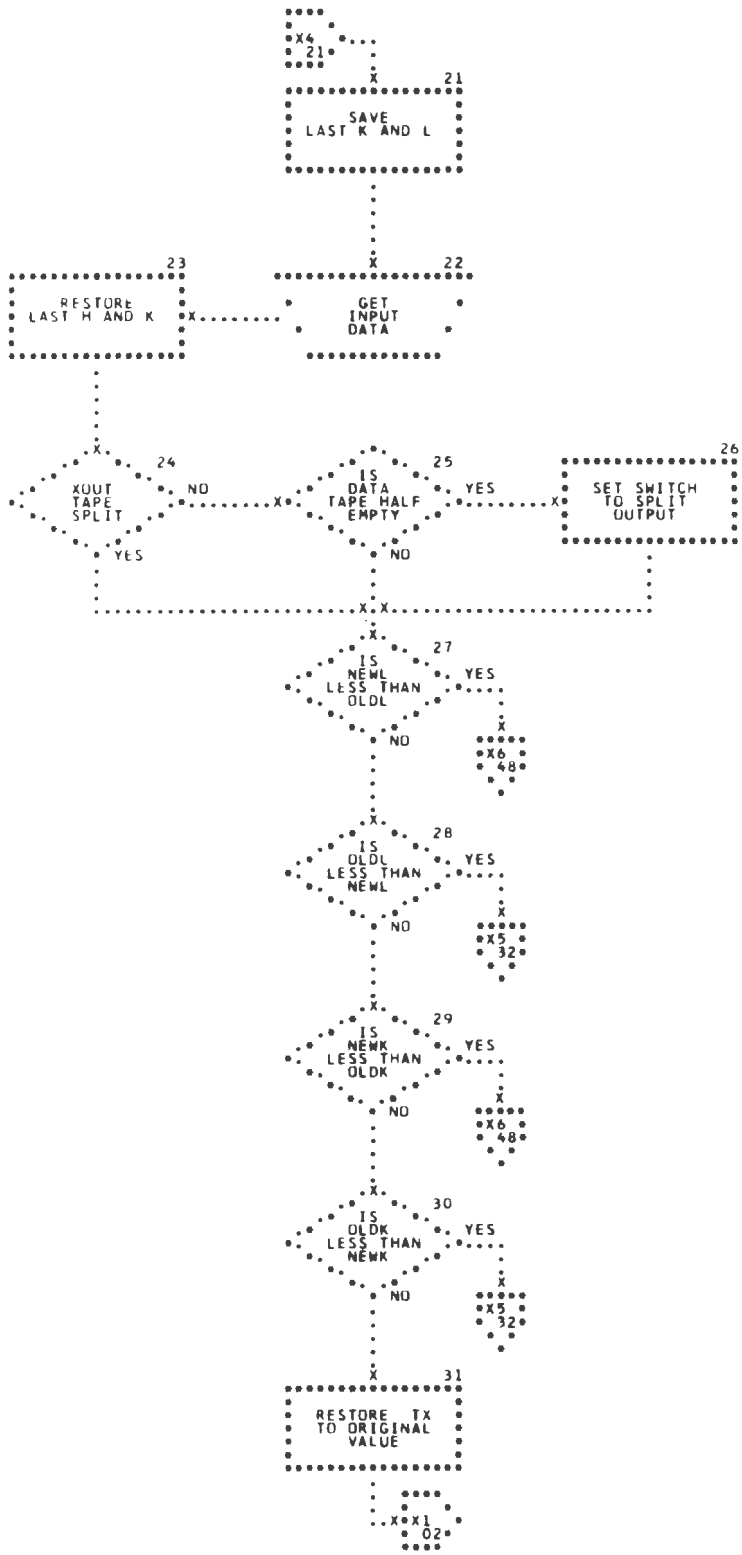
Y005 This is a branch switch that is off (NOP) if output is waiting. See block X007.

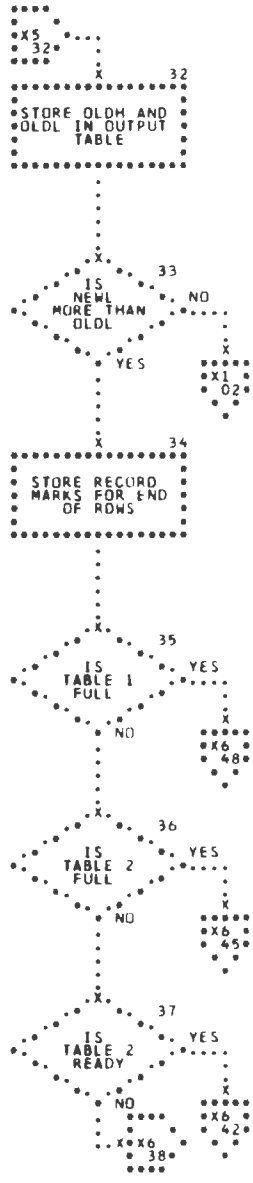
Y006 Is the output tape busy? See block X008.

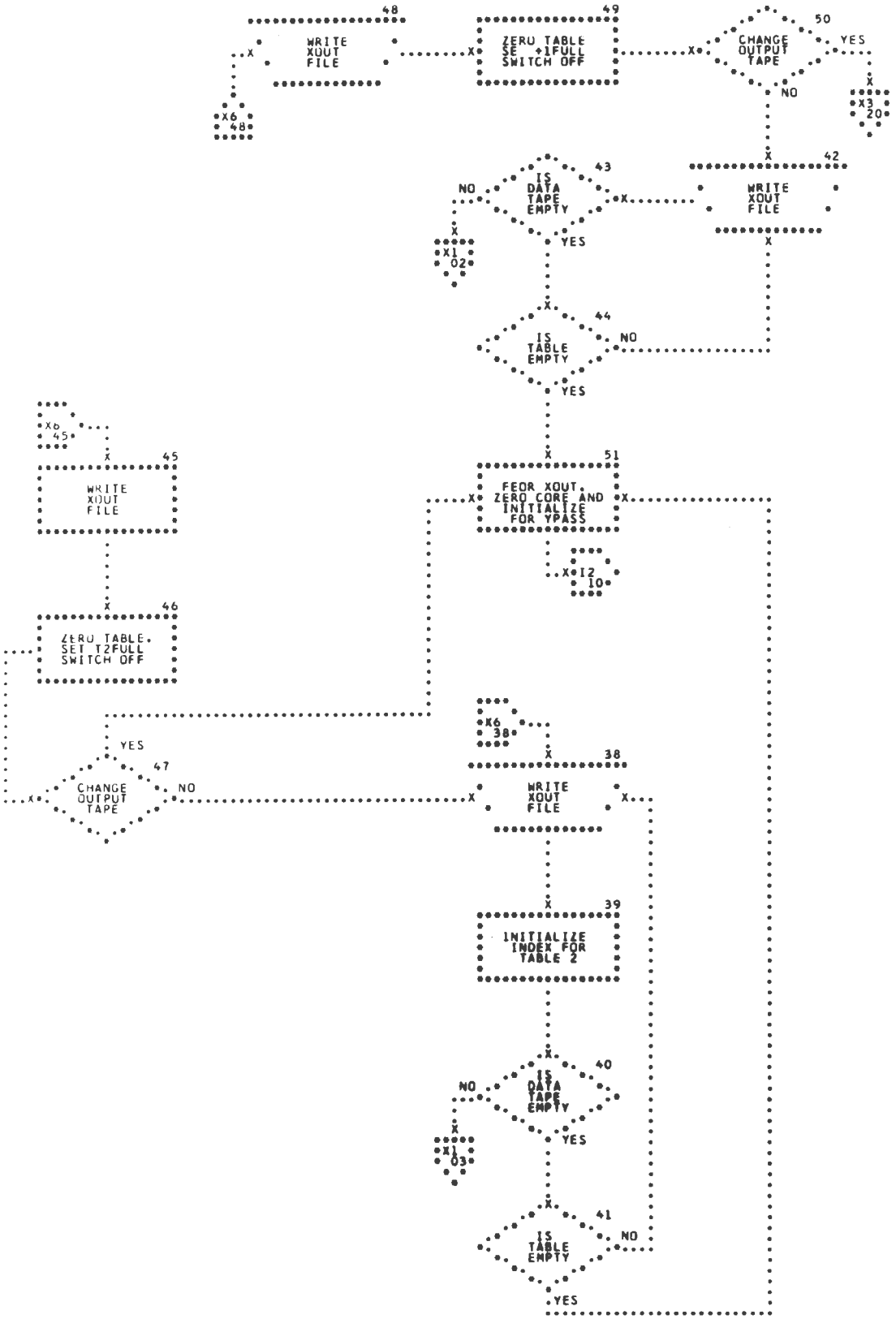

00
START
01
GET INPUT DATA
02
LOAD INDEX TX FOR FIRST STORAGE TABLE
X1 02
03
LOAD ACCUMULATOR 3 WITH (F)
X1 03
04
SCALE FACTOR
NO
YES
05
SHIFT RIGHT OR LEFT
06
LOAD INDEX INDX WITH (4)
BUFFERED OUTPUT
07
IS OUTPUT WAITING
NO
YES
X3 18
08
IS OUT CHANNEL BUSY
YES
NO
X3 18
09
IS THIS TABLE 1 OUTPUT
NO
YES
X2 14
X2 10


X2
10
10
WRITE
XOUT
FILE
11
IS THIS
LAST
RECORD
NO
YES
GO
TO CHART X3,
BLOCK 20
12
ZERO TABLE,
SET T1FULL SW
OFF, SET WAITIN
B TO NO.
13
CHANGE
OUTPUT
TAPE
NO
YES
GO
TO CHART X1,
BLOCK 02
X2
14
14
WRITE
XOUT
FILE
15
IS THIS
LAST
RECORD
NO
YES
16
ZERO TABLE,
SET T2FULL SW
OFF, SET WAITIN
B TO NO.
17
CHANGE
OUTPUT
TAPE
NO
YES


X3
18
18
INITIALIZE FOR CORRECT SINCOS TABLE ENTRY
19
SET BRANCH TO INITIALIZE FOR TX
MO
01
X3
20
20
FEOR XOUT
X4
40
X4
45


X4
21
21
SAVE
LAST K AND L
23
RESTORE
LAST H AND K
22
GET
INPUT
DATA
24
XOUT
TAPE
SPLIT
NO
YES
25
IS
DATA
TAPE HALF
EMPTY
YES
NO
26
SET SWITCH
TO SPLIT
OUTPUT
27
IS
NEWL
LESS THAN
OLDL
YES
NO
X6
48
28
IS
OLDL
LESS THAN
NEWL
YES
NO
X5
32
29
IS
NEWK
LESS THAN
OLDK
YES
NO
X6
48
30
IS
OLDK
LESS THAN
NEWK
YES
NO
X5
32
31
RESTORE IX
TO ORIGINAL
VALUE
X1
02


X5
32
32
STORE OLDH AND OLDL IN OUTPUT TABLE
33
IS NEWL MORE THAN OLDL
NO
YES
X1
02
34
STORE RECORD MARKS FOR END OF ROWS
35
IS TABLE 1 FULL
YES
NO
X6
48
36
IS TABLE 2 FULL
YES
NO
X6
45
37
IS TABLE 2 READY
YES
NO
X6
42
X6
38


48
WRITE XOUT FILE
49
ZERO TABLE SE +1FULL SWITCH OFF
50
CHANGE OUTPUT TAPE
YES
NO
X6 48
X3 20
42
WRITE XOUT FILE
43
IS DATA TAPE EMPTY
NO
YES
X1 02
44
IS TABLE EMPTY
NO
YES
X6 45
45
WRITE XOUT FILE
51
FEOR XOUT, ZERO CORE AND INITIALIZE FOR YPASS
I2 10
46
ZERO TABLE, SET T2FULL SWITCH OFF
X6 38
38
WRITE XOUT FILE
YES
47
CHANGE OUTPUT TAPE
NO
39
INITIALIZE INDEX FOR TABLE 2
40
IS DATA TAPE EMPTY
NO
YES
X1 03
41
IS TABLE EMPTY
NO
YES

BLOCK #

Y007 Output is waiting - is it Table #1? See block X009.

Y008 Write out one record from Table #1.

Y009 Was this the last record in Table #1?

Y010 Zero table and initialize for new summation for use when Table #2 is full. Turn on the branch switch to bypass checks.

Y011 If output is to be split here CHNGTPE is on. CHNGTPE is triggered by the EOF of the first input tape.

Y012 Same for Table #2 as block Y008.

Y013 Same for Table #2 as block Y009.

Y014 Same for Table #2 as block Y010.

Y015 Same for Table #2 as block Y011. The output tape can be signaled from either table.

Y016 YLPX is decremented by 3 for each entry. After the fourth entry YLPX is re-initialized.

Y017 INDX is incremented to refer to the sine table or the cosine table.

Y018 Set up YLPX to begin with four new entries.

Y019 The next entry is 5 asterisks signaling the end of a X sum table.

Y020 Set up table for output. Put the L value for this table directly behind the 5 asterisks.

Y021 If next entry is a record mark we are at the end of a record but not the end of a table.

Y022 Next entry is data so load INDX with the new K and add 5 to YINA to refer to the first of four entries. Subtract 1 from TX to refer to correct output entry.

Y023 Add 1 to TX as new record must not be summed with last one.

Y024 Get input record.

Y025 See Block Y003.

BLOCK #

Y026 Set OUTDONEY off (NOP) because output is waiting.

Y027 OUTDONEY may have already been off- is Table #2 empty?

Y028 Is Table #1 empty?

Y029 Last output table is empty - if T2 is on Table #2 is the new table.

Y030 Write out one record from Table #1.

Y031 Initialize for a new summation in Table #2.

Y032 Write out one record from Table #2.

Y033 Initialize for a new summation in Table #1.

Y034 Table #1 is not empty. Write out one record.

Y035 Is Table #1 empty?

Y036 Table is empty; zero table and go directly to Block Y032 since Table #2 must be the new table.

Y037 See Block Y011.

Y038 Same for Table #2 as Block Y034.

Y039 Same for Table #2 as Block Y035.

Y040 Same for Table #2 as Block Y036.

Y041 See Block Y011.

Y042 FEOR will alternate the Y sum output tapes. Go back to BLX-1 to continue program.

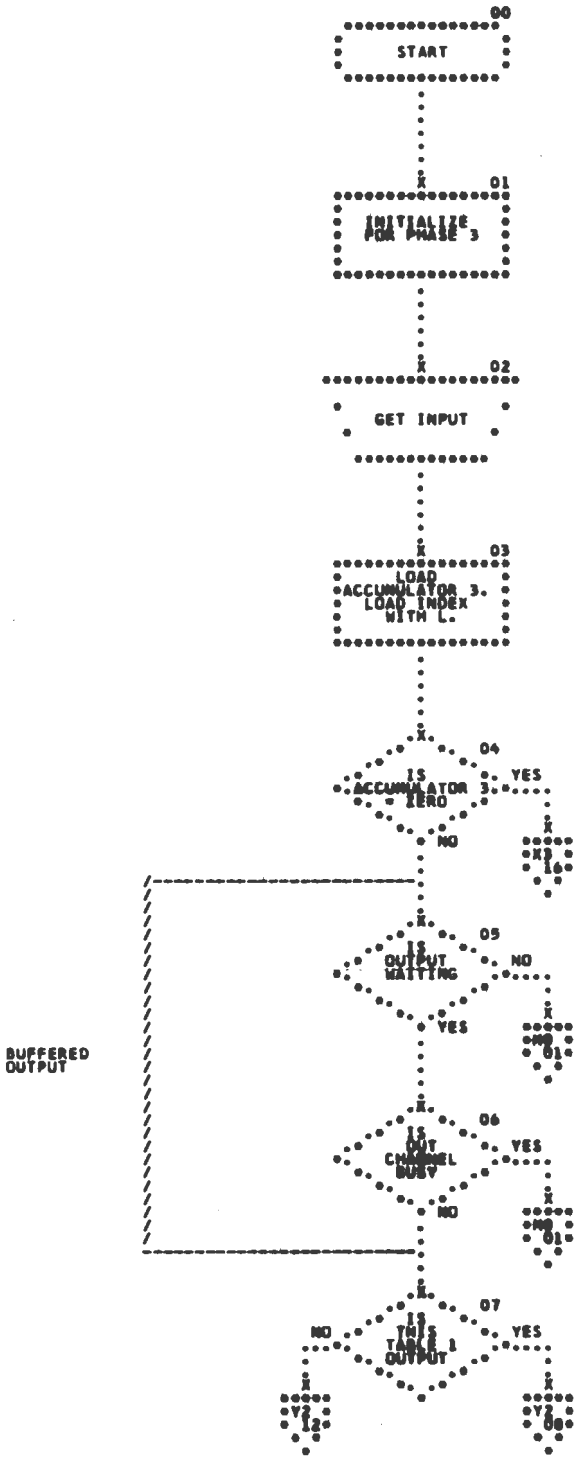
4H. Z-Summation Phase (#4) Flow Diagrams

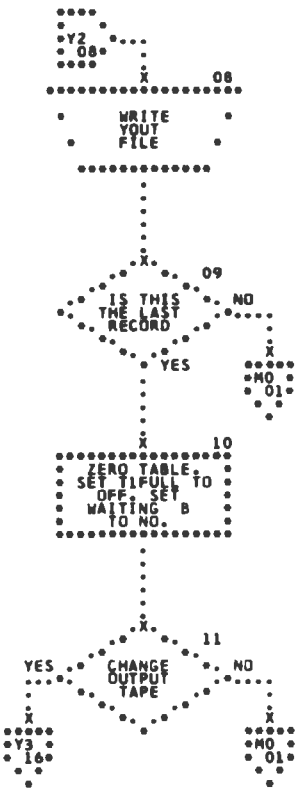
BLOCK #

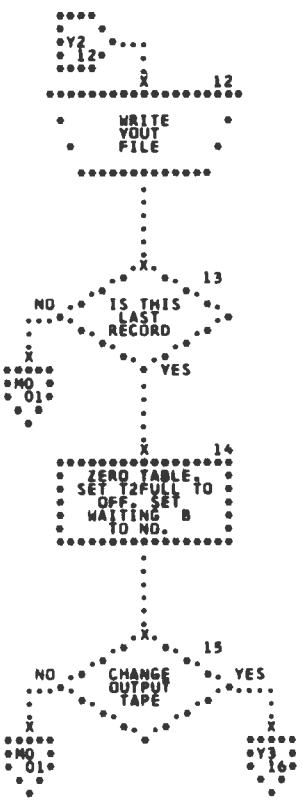
Z001 Get input from Y summation output.

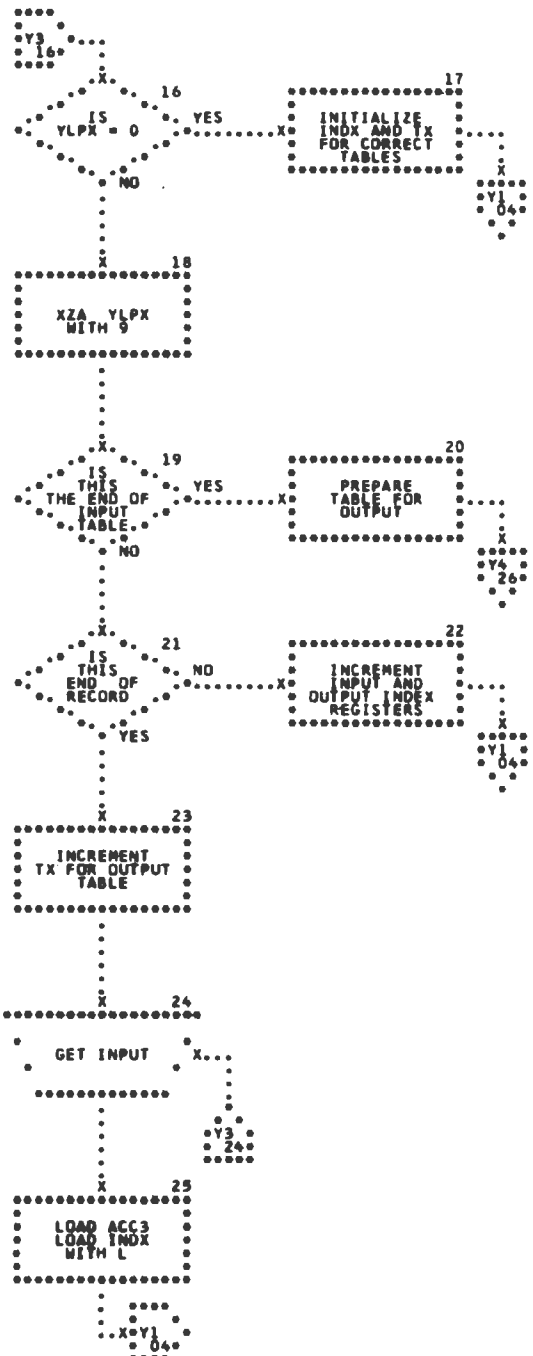
Z002 Pick up first word of record.

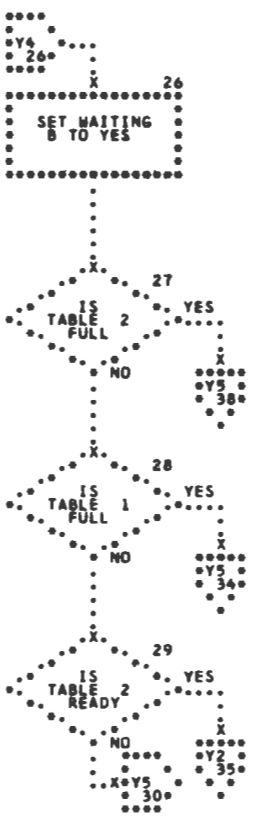
Z003 If first word is five asterisks the new L value is the next word.

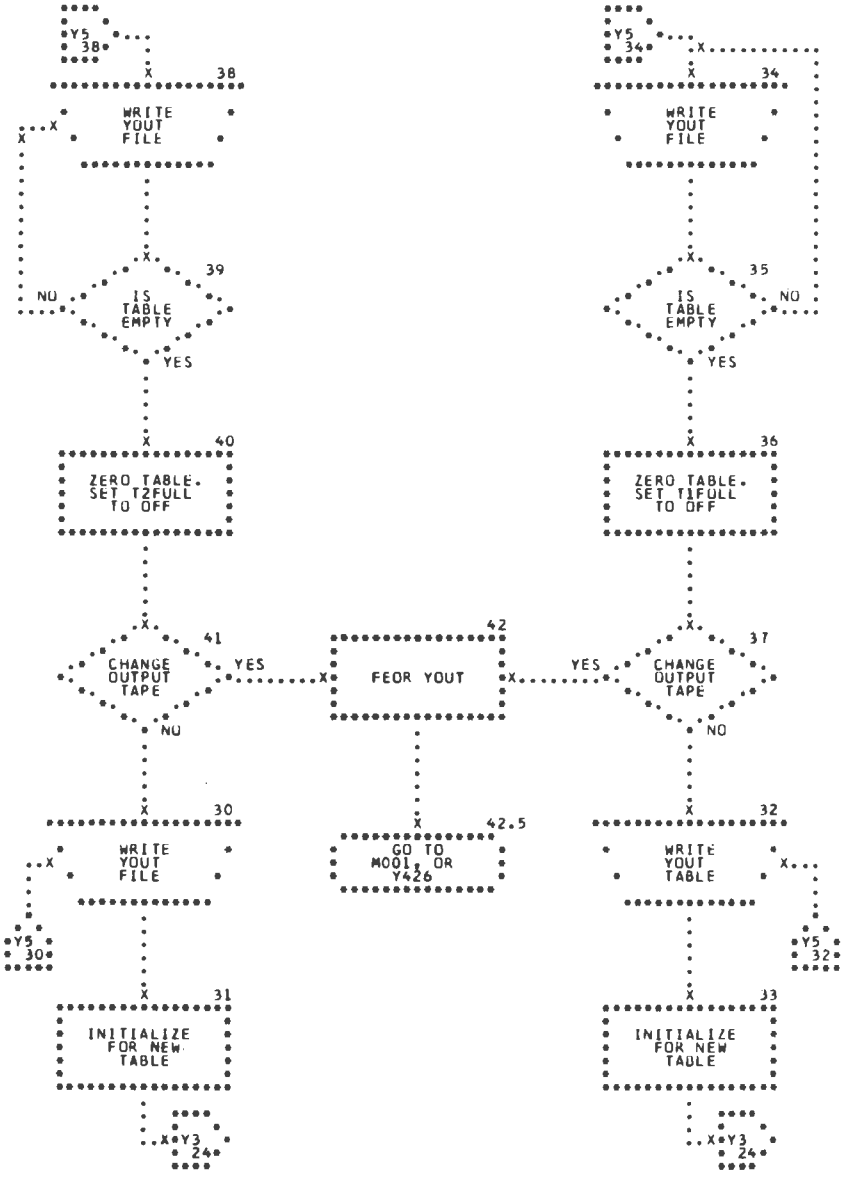

00
START
01
INITIALIZE FOR PHASE 3
02
GET INPUT
03
LOAD ACCUMULATOR 3. LOAD INDEX WITH L.
04
IS ACCUMULATOR 3 = ZERO
YES
NO
X3 16
05
IS OUTPUT WAITING
NO
YES
M0 01
BUFFERED OUTPUT
06
IS OUT CHANNEL BUSY
YES
NO
M0 01
07
IS THIS TABLE 1 OUTPUT
NO
YES
Y2 12
Y2 08


Y2
08
08
WRITE
YOUT
FILE
09
IS THIS
THE LAST
RECORD
NO
YES
MO
01
10
ZERO TABLE,
SET T1FULL TO
OFF, SET
WAITING B
TO NO.
11
CHANGE
OUTPUT
TAPE
YES
NO
Y3
16
MO
01


Y2
12
12
WRITE
YOUT
FILE
13
IS THIS
LAST
RECORD
NO
YES
MO
01
14
ZERO TABLE,
SET T2FULL TO
OFF, SET
WAITING B
TO NO.
15
CHANGE
OUTPUT
TAPE
NO
YES
MO
01
Y3
16


Y3
16
16
IS
YLPX = 0
YES
NO
17
INITIALIZE
INDX AND TX
FOR CORRECT
TABLES
Y1
04
18
XZA YLPX
WITH 9
19
IS
THIS
THE END OF
INPUT
TABLE
YES
NO
20
PREPARE
TABLE FOR
OUTPUT
Y2
26
21
IS
THIS
END OF
RECORD
NO
YES
22
INCREMENT
INPUT AND
OUTPUT INDEX
REGISTERS
Y1
04
23
INCREMENT
TX FOR OUTPUT
TABLE
24
GET INPUT
Y3
24
25
LOAD ACC3
LOAD INDX
WITH L
Y1
04


Y4
26
X
26
SET WAITING
B TO YES
X
27
IS
TABLE 2
FULL
YES
NO
X
Y5
38
X
28
IS
TABLE 1
FULL
YES
NO
X
Y5
34
X
29
IS
TABLE 2
READY
YES
NO
X
Y2
35
X
Y5
30


Y5
38
38
WRITE
YOUT
FILE
39
IS
TABLE
EMPTY
NO
YES
40
ZERO TABLE.
SET T2FULL
TO OFF
41
CHANGE
OUTPUT
TAPE
YES
NO
42
FEOR YOUT
42.5
GO TO
M001, OR
Y426
30
WRITE
YOUT
FILE
Y5
30
31
INITIALIZE
FOR NEW
TABLE
Y3
24
Y5
34
34
WRITE
YOUT
FILE
35
IS
TABLE
EMPTY
NO
YES
36
ZERO TABLE.
SET T1FULL
TO OFF
37
CHANGE
OUTPUT
TAPE
YES
NO
32
WRITE
YOUT
TABLE
Y5
32
33
INITIALIZE
FOR NEW
TABLE
Y3
24

BLOCK #

Z004 Load INDX with L and zero BLDCNT so that TX is loaded with RDW for first output record.

Z005 Load ACC3 with third word, which is first input. Also increment ZINA to refer to first valid input word. Turn off ZBLK as for first word we do not check to see if ACC3 = 0 so assume not. Next word must be multiplied by $\frac{2\pi 1Z}{160}$ so set SNBR on to to increment INDX for next word. Go to mult. routine.

Z006 This is a valid input word since only first record of each block has asterisks. Load TX to refer to next output record. Add one to BLDCNT so that TX will be loaded correctly next time.

Z007 Is input word zero? If so do not multiply; turn off branch to multiply routine.

Z008 Turn off switch that will allow branch to multiply routine.

Z009 Was last word multiplied by a sine value?

Z010 No, so this one must be. Set switch to go to Block Z011 for next word.

Z011 Yes, so subtract difference between table entries that refer to sine and cosine. Turn off switch so that next word does not go to this block.

Z012 If ACC3 = 0 switch is turned off in Block Z008.

Z013 Branch is turned on for next word. A branch is then made to ZLOOP which is entry point from multiply routine. ACC3 is loaded with new input word. Input index register is incremented.

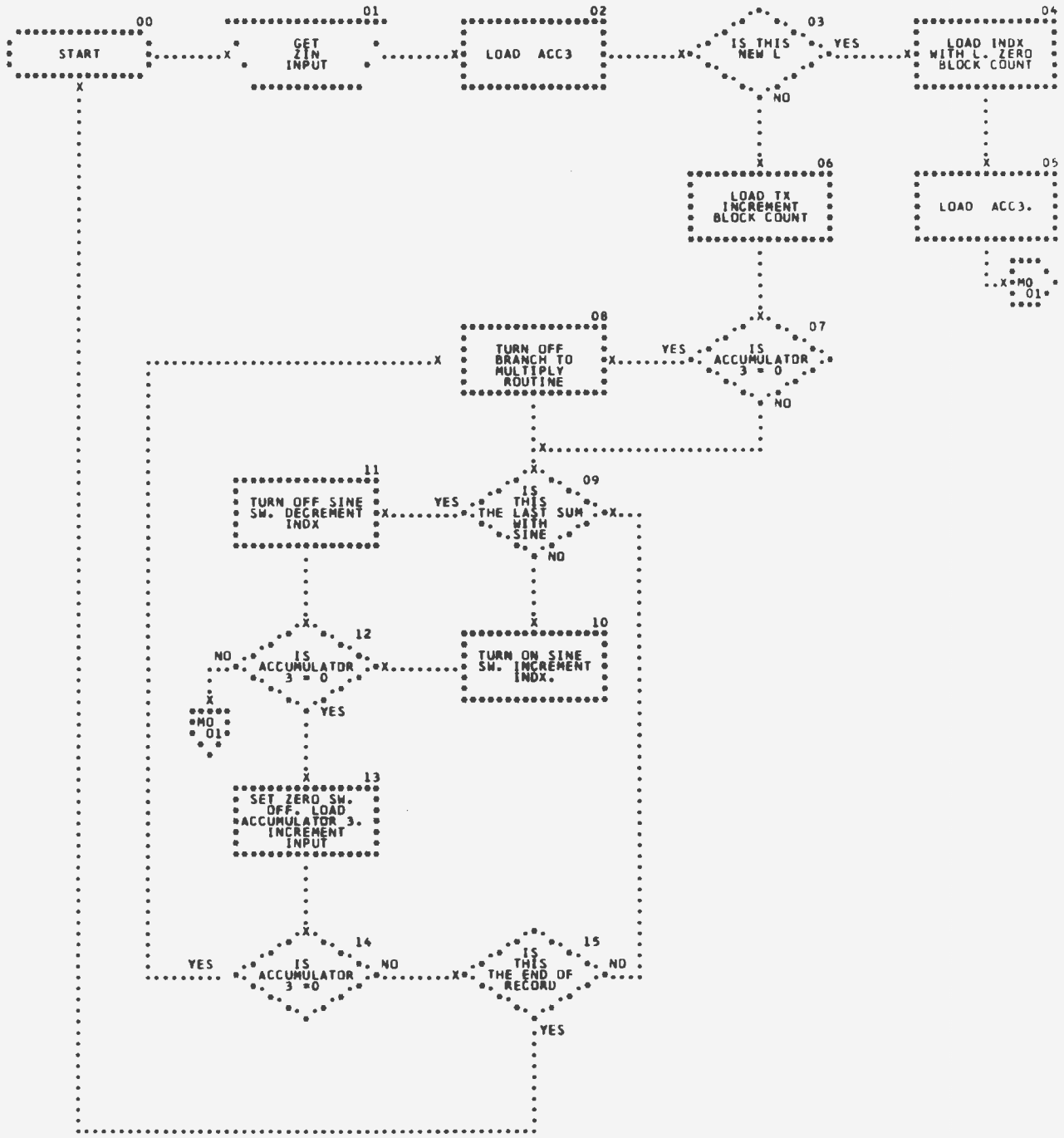
Z014 If ACC3 = 0 do not multiply.

Z015 ACC3 is not zero - is it loaded with a record mark? If so go for new record as record mark signals end of record.

4I. Multiplication Routine Flow Diagrams

BLOCK #

M001 This block is reached from any one of three places, one each in all three summations. Since this routine is used for all three summations, it must be modified before each summation. This is done just prior to branching to the correct summation phase from the initialization phase.


00
START
01
GET ZIN INPUT
02
LOAD ACC3
03
IS THIS NEW L
YES
NO
04
LOAD INDX WITH L. ZERO BLOCK COUNT
05
LOAD ACC3.
MO 01
06
LOAD TX INCREMENT BLOCK COUNT
07
IS ACCUMULATOR 3 = 0
YES
NO
08
TURN OFF BRANCH TO MULTIPLY ROUTINE
09
IS THIS THE LAST SUM WITH SINE
YES
NO
10
TURN ON SINE SW. INCREMENT INDX.
11
TURN OFF SINE SW. DECREMENT INDX
12
IS ACCUMULATOR 3 = 0
NO
YES
MO 01
13
SET ZERO SW. OFF. LOAD ACCUMULATOR 3. INCREMENT INPUT
14
IS ACCUMULATOR 3 =0
YES
NO
15
IS THIS THE END OF RECORD
NO
YES

BLOCK #

For each of 21 values of, say, x the cosine-sine routine supplies MAHX+1 values of $\cos \frac{2\pi hx}{160}$ and MAXH+1 values of $\sin \frac{2\pi hx}{160}$.

Each one of the values corresponds to a value of h in increasing sequence from h = 0 to h = MAXH. Then the multiplication routine only has to refer to the first value, i.e., h = 0 for every one of the 21 values of x and by loading INDX correctly the correct sine or cosine value will be obtained.

M002 This part of the routine is also modified for each phase. TX is always loaded with an RDW referring to one of the summation records. As an example, consider the X summation.

Provisions have been made to accept a maximum of 50 values of H, K, or L. Then, since each value of K requires 5 words in a summation record, 250 words must be provided for every value of X. Therefore, the routine is modified so that every AS1 is 250 locations above the preceding one (starting with zero). By loading TX with the RDW for the first record in either table each AS1 refers to the first word of the Xth record where AS1 is the ith AS1.

If less than 20 points are required, the branch at the end of the routine (TT20A+1) is moved up to be directly below the last needed AS1.

4J. Edit Routine Flow Diagrams

BLOCK #

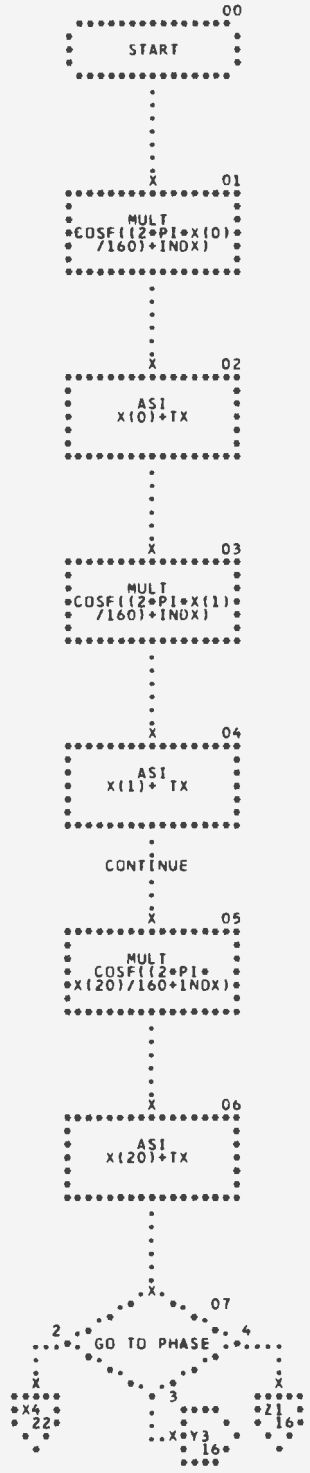
E001 XSTART, YSTART, and ZSTART are edited and entered into label.

E002 Label is written on tape.

E003 TSIZE2 is loaded with ZPTS as this is the number of tables that will be written out.

E004 TSIZE1 is set up with the end of the record in (2, 5) and the end of the edit area in (6, 9). If a BCX is given later, we can determine whether the full edit area should be printed. An RDW, TSIZEDC1, is created for blanking-out if needed.

E005 YPTS determines the number of records in one table to TSIZE is loaded accordingly.


00
START
01
MULT
COSF((2*PI*X(0)
/160)+INDX)
02
ASI
X(0)+TX
03
MULT
COSF((2*PI*X(1)
/160)+INDX)
04
ASI
X(1)+TX
CONTINUE
05
MULT
COSF((2*PI*
X(20)/160+INDX)
06
ASI
X(20)+TX
07
2
GO TO PHASE
4
3
X4
22
Y3
16
Z1
16

BLOCK #

E006 The beginning of each table (each page of output) is located 21 RDWs below the beginning of the last table. We will use this index register to pick up each RDW.

E007 Do not increment indexing portions until needed.

E008 Set indexing portion to be 20 greater than non-indexing since BIX has already incremented once.

E009 Load WORKX for editing of correct record.

E010 Use EDMOV to edit one record.

E011 Index register mentioned in Block E004 is now checked to see if part of edit should be blanked-out.

E012 Blank-out part not needed. It is filled with alpha blanks.

E013 Output edited record.

E014 Index register mentioned in Block E005 is now tested to see if this is the end of a table. WORKX1 cannot be used for this purpose since YPTS may not be exactly 20.

E015 Increment WORKX1 so that BIX in Block E019 fails.

E016 Increment Z coordinate as new table must be started.

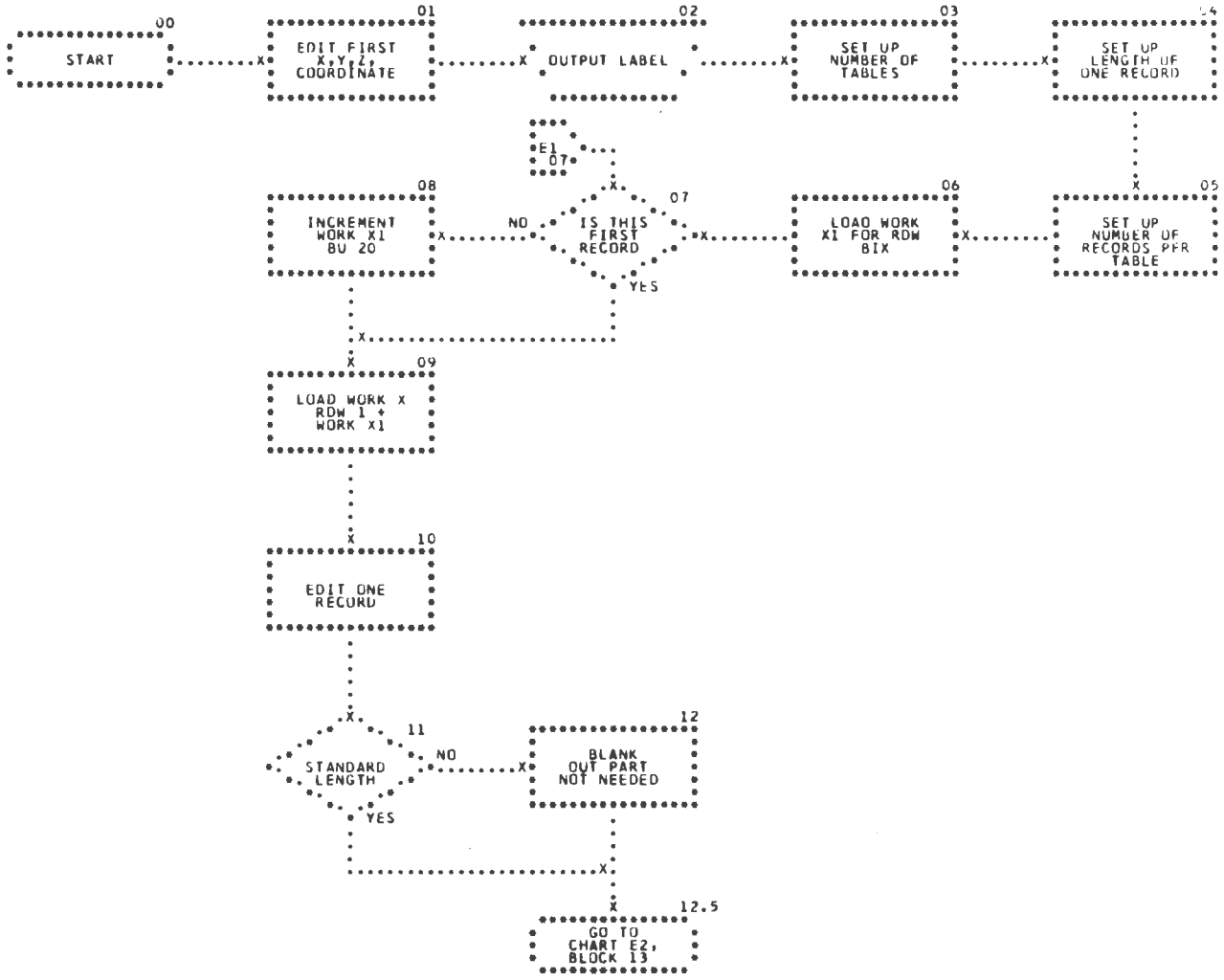
E017 If we are finished do not write out label.

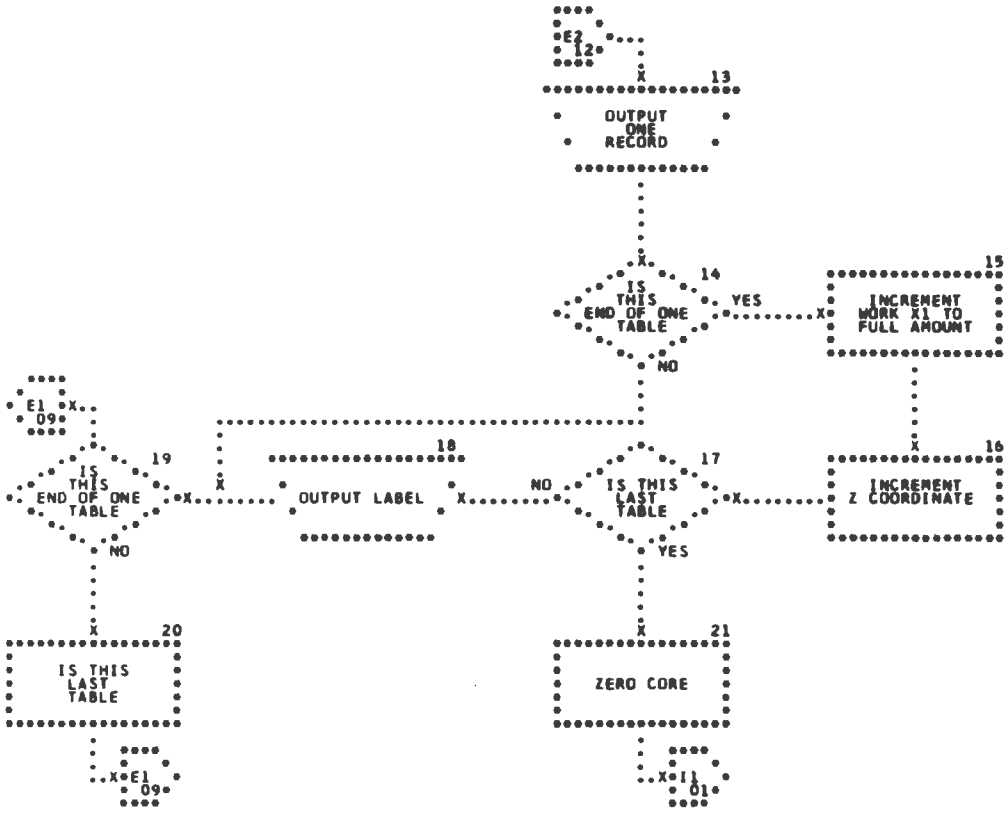
E018 Output label.

E019 If one table is finished Block E015 has incremented WORKX1 so that the BIX fails.

E020 If this is last table do not go back. Actually this BIX never fails since Block E017 fails first.

E021 All of the core below the output area is zeroed and the last control card is moved to an area for comparison purposes. If only one output tape was used in the X and Y summations ONEXTP will be on so go to open both files. The files were closed instead of using FEOR so that the tape units would not be rotated.


00
START
01
EDIT FIRST X,Y,Z, COORDINATE
02
OUTPUT LABEL
03
SET UP NUMBER OF TABLES
04
SET UP LENGTH OF ONE RECORD
05
SET UP NUMBER OF RECORDS PER TABLE
06
LOAD WORK X1 FOR RDW BIX
07
IS THIS FIRST RECORD
NO
YES
E1
07
08
INCREMENT WORK X1 BY 20
09
LOAD WORK X RDW 1 + WORK X1
10
EDIT ONE RECORD
11
STANDARD LENGTH
NO
YES
12
BLANK OUT PART NOT NEEDED
12.5
GO TO CHART E2, BLOCK 13


E2
12
13
OUTPUT ONE RECORD
14
IS THIS END OF ONE TABLE
YES
NO
15
INCREMENT WORK X1 TO FULL AMOUNT
16
INCREMENT Z COORDINATE
17
IS THIS LAST TABLE
NO
YES
18
OUTPUT LABEL
19
IS THIS END OF ONE TABLE
NO
E1
09
20
IS THIS LAST TABLE
E1
09
21
ZERO CORE
I1
01

4K. Sine-Cosine Table

See Fig. 1 for layout of the table. This table is generated for the first X summation and, thereafter, every time a summation needs a different table of values.

The values in the table are taken from a table consisting of 161 values each of the sine and cosine going from 0 to 2π in even increments. By calculating the length of the cosine table we need only add this value to INDX to obtain the correct sine. The cosine-sine routine calculates and stores this value in various instructions throughout the program.

Although all 21 points, i. e., X_0 to X_{20}, are not always needed, the tables are generated because of convenience and simplicity.

4L. X-Summation Tables

Only one table(Fig. 2) will be shown since both tables are identical.

Note that the value of L does not change in any one table although K changes for each block of four entries. Since L does not change in any one table whenever a new value of L is reached in the data input the table must be closed by writing record marks and the five asterisks. Then while the table is being written on tape a new table is being constructed in the other area available.

The records in one table are always of the same length and may contain many blank entries. This makes it imperative to check each Y sum input entry before multiplying.

Upon reaching an input record (Y sum) having five asterisks the program expects the next record to contain a new value of L.

X VALUE	H VALUE	TABLE VALUE
X_0	0	$\cos\left(\frac{2\pi X0}{160}(0)\right)$
X_0	1	$\cos\left(\frac{2\pi X0}{160}(0)\right)$
.	.	.
.	.	.
.	.	.
X_0	MAXH	$\cos\left(\frac{2\pi X_0}{160}(0)\right)$
X_1	0	$\cos\left(\frac{2\pi X1}{160}(0)\right)$
X_1	1	$\cos\left(\frac{2\pi X1}{160}(1)\right)$
.	.	.
.	.	.
.	.	.
X_{20}	0	$\cos\left(\frac{2\pi X_{20}}{160}(0)\right)$
.	.	.
.	.	.
.	.	.
X_{20}	MAXH	$\cos\left(\frac{2\pi X_{20}}{160}(MAXH)\right)$
X_0	0	$\sin\left(\frac{2\pi X_0}{160}(0)\right)$
.	.	.
.	.	.
.	.	.
X_{20}	0	$\sin\left(\frac{2\pi X_{20}}{160}(0)\right)$
.	.	.
.	.	.
.	.	.
X_{20}	MAXH	$\sin\left(\frac{2\pi X_{20}}{160}(MAXH)\right)$

Fig. 1

ENTRY	TABLE VALUE	X VALUE
Record #1		
1	SCC + CCC	X_0
2	SSC + CSC	X_0
3	SSS + CSS	X_0
4	SCS + CCS	X_0
5	K_1 L_1	X_0
.	.	.
.	.	.
.	.	.
n	SCC + CCC	X_0
n+1	SSC + CSC	X_0
n+2	SSC + CSC	X_0
n+3	SCS + CCS	X_0
n+4	K_{max} L_1	X_0
n+5	RECORD MARK	X_0
Record #2		
1	SCC + CCC	X_1
2	SSC + CSC	X_1
3	SSS + CSS	X_1
4	SCS + CCS	X_1
5	K_1 L_1	
.	.	.
.	.	.
.	.	.
Record #21		
1	SCC + CCC	X_{20}
.	.	.
.	.	.
.	.	.
n+4	K_{max} L_1	X_{20}
n+5	*****	
n+6	RECORD MARK	

Fig. 2

A maximum of 50 values of K can be entered into one record. This means each record is a maximum of 250 words plus 2 for the record mark and five asterisks.

4M. Y-Summation Tables

Only one table (Fig. 3) will be shown since both tables are identical.

Note that L remains fixed for every record in one table. A new value of L is signaled by the five asterisks in front of a record. At this time the table is written on tape and a new table is built using the values corresponding to the new L.

ACKNOWLEDGMENTS

I would like to thank D. R. Fitzwater for technical assistance, J. Jackobs for providing me with several data tapes for testing purposes, and C. Runge for his help in locating errors and making modifications.

ENTRY	TABLE VALUE	X VALUE	Y VALUE
Record #1			
1	*****		
2	L_1		
3	SCC+CCC+SSC+CSC	X_0	Y_0
4	SSS+CSS+SCS+CCS	X_0	Y_0
5	SCC+CCC+SSC+CSC	X_1	Y_0
6	SSS+CSS+SCS+CCS	X_1	Y_0
.	.	.	.
.	.	.	.
.	RECORD MARK	.	.
Record #21			
1	SCC+CCC+SSC+CSC	X_0	Y_{20}
2	SSS+CSS+SCS+CCS	X_0	Y_{20}
.	.	.	.
.	.	.	.
.	.	.	.
41	SCC+CCC+SSC+CSC	X_{20}	Y_{20}
42	SSS+CSS+SCS+CCS	X_{20}	Y_{20}
43	RECORD		

Fig. 3